ECE 5256

Matlab project: Read and display an image and Affine transformation

1 Determine a process where you can acquire an image and use Matlab commands to get that image into Matlab. Make sure you can read in gray-scale images.

See if you can use the command *imread*. You can always use the following for reading and writing .raw images, those without a header. Below is the code for reading in a 400H x 200V image in .raw format named ***image.raw***. Note that a, .raw, .bin, or no extension may be used, but the extension must match the file. The variable ***a*** contains the image. Note that the last line transposes the image. If you created a file image using Matlab, the last line may be needed because of the way the data is stored.

```
fid=fopen('image.raw');
a=fread(fid,[400,200],'type');
fclose(fid);
a=a';
```

Where *type* is a variable type such as:

schar - 8-bit signed character,
uchar - 8-bit unsigned character,
uint16 - 16-bit unsigned integer,
int16 - 16-bit signed integer,
float - 32-bit floating point.

You can write an image to a file in a format without a header. It is the same as the .raw format. (or .bin) The following code puts the contents of the variable ***result*** into a file named ***file***. The data can be stored as different types.

```
fid=fopen('file','wb');
fwrite(fid,result,'type');
```

2 Read in an image and display it.
Depending on the version of Matlab it will display 32, 64 or some other levels on your screen. An image can be displayed using the command ***image(a)***, where ***a*** is the variable that contains the image. The ***image*** command usually assigns 32 gray levels between 0 and 31. Values greater than 31 are mapped to white, and below 0 are mapped to black. This can be inconvenient so the ***imagesc*** command may be used. This command automatically scales the display. The largest value of the image is set to white, and the lowest is set to black with a total of 32 gray levels from white to black.

In addition, you can set the scaling to whatever you want. The example below scales an image between 0 and x. Anything less than x gets mapped to black, and anything greater than x gets mapped to white.

```
clims=[0,x];
imagesc(image, clims)
```

Note that the value of the variable doesn't change in any of these examples, only the display on the screen.

The ***colormap*** command allows various look up tables to be used. To display an image in grayscale, use the command ***colormap(gray)***.

Make yourself familiar with axis commands such as: “image,” “off,” etc.

3 Read in an image and scale the intensity values from 0 – 255.

Add noise to the image with a mean and variance of 20 using the randn command.

Take the average of *N* noisy versions of the original (using independent noise samples!) for values of *N* = 0, 5, and 10. and compare the result to the original image using the MSE. Does the MSE decrease with *N* or the sqrt(*N*)?

The MSE is a scalar and is calculated as $\sum\sum$(original image – averaged image)^2

4 Affine transformation. Read in an image, and rotate and scale it by 35 degrees and a scale of 0.7 in all directions.

Turn in the following:

1. A brief write-up of what you did.
2. An image displayed with 32 and 255 intensity levels.
3. An image displayed with two different colormaps.
4. MSE vs. N for noisy images
5. Result of affine transformation.